

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

2014

Jan Vožický

Zadání bakalářské práce

Student: **Jan Vožický**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: CATHEDRAL Software, s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadáných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadáných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Gaura**

Konzultant bakalářské práce: Ing. Josef Šustr

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



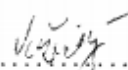
doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 7.května 2014

.....

Chtěl bych poděkovat Ing. Josefu Šustrovi za odborné vedení, trpělivost a ochotu, kterou mi v průběhu zpracování bakalářské práce věnoval. A dále bych rád poděkoval panu Ing Janu Gaurovi za pomoc při zpracování práce.

Abstrakt

Obsahem této práce je průběh mé odborné praxe absolvované ve firmě Cathedral Software, s.r.o. Popisuji zde zaměření firmy a úkoly, které jsem řešil. U úkolů uvádím jejich zadání a následně můj postup při řešení těchto úkolů.

Poté shrnuji teoretické a praktické znalosti a dovednosti získané během studia, které jsem při plnění těchto úkolů využil, stejně tak jako znalosti získané v průběhu praxe. Na závěr hodnotím celý průběh praxe.

Klíčová slova: Odborná praxe, Cathedral Software, Python, Web2Py, Django

Abstract

Content of this bachelor thesis is process of my professional practice, which i passed in Cathedral Software, s.r.o. Here i present focus of the company and projects which I've been working on. For projects I mention their task and my procedure to resolve these projects.

Thereafter i sum up teoretical and practical knowledge and skills I gained during my studies, which I used at work, as well as skills gained from my professional practice. In the end I evaluate whole process of my practice.

Keywords: Professional practice, Cathedral Software, Python, Web2Py, Django

Seznam použitých zkratk a symbolů

CSS	– Cascading Style Sheets
GUI	– Graphical user interface
HTML	– HyperText Markup Language
MSSQL	– Microsoft Structured Query Language Server
MVC	– Model-view-controller
ORM	– Object-relational mapping
URL	– Uniform resource locator

Seznam obrázků

1	Logo firmy	4
2	Závislosti mezi MVC	8

Obsah

1	Úvod	3
2	Informace o praxi	4
2.1	Popis odborného zaměření firmy	4
2.2	Popis pracovního zařazení studenta	4
3	Seznam úkolů	5
3.1	Možnosti web rozhraní pro přístup k datům s využitím jazyka Python . .	5
3.2	Tvorba webového rozhraní pro přístup k datům pomocí frameworku Web2Py	5
3.3	Tvorba webového rozhraní pro přístup k datům pomocí frameworku Django	5
4	Zvolený postup řešení	6
4.1	Možnosti web rozhraní pro přístup k datům s využitím jazyka Python . .	6
4.2	Tvorba webového rozhraní pro přístup k datům pomocí frameworku Web2Py	9
4.3	Tvorba webového rozhraní pro přístup k datům pomocí frameworku Django	12
5	Uplatněné teoretické a praktické znalosti	14
6	Scházející znalosti a dovednosti	15
7	Závěr	16
8	Reference	17

1 Úvod

Cílem této práce je popsat průběh praxe, kterou jsem vykonával ve firmě Cathedral Software, s.r.o. V následujícím textu popisuji zaměření firmy a mé zařazení v ní. Dále popisuji zadání úkolů, které jsem musel v průběhu praxe vykonat, zvolený postup a technologie a v neposlední řadě problémy, na které jsem při řešení narazil. V následující části popisuji praktické znalosti a dovednosti, které jsem získal během studia a využil je v praxi a naopak vědomosti, které mi chyběly a musel jsem se je v průběhu naučit. Na závěr shrnuji celý průběh praxe a popisuji, co mi tato zkušenost dala.

2 Informace o praxi

2.1 Popis odborného zaměření firmy

Společnost Cathedral Software [1] nabízí zákazníkům komplexní služby v oblasti informačních a komunikačních technologií. Soustředí se hlavně na vývoj informačních systémů. Především se jedná o informační systém pro obchodní a výrobní firmy ArisCAT. Tento software slouží ke zpracování nabídek a konstrukci výrobků stínící techniky, u které existuje velká variabilita ve zpracování a použitém materiálu. Dále se firma zabývá vývojem a distribucí integrovaného modulu pro informační systémy Helios Orange. Ten pomáhá managementu v řízení všech potřebných oblastí. Poskytuje dokonalý přehled o situaci na trhu i uvnitř podniku. Cathedral Software se také zabývá tvorbou softwaru na zakázku, prodejem značkové výpočetní techniky a v neposlední řadě firma k poskytnutým službám nabízí školení, konzultaci a poradenství.



Obrázek 1: Logo firmy

2.2 Popis pracovního zařazení studenta

Po zaškolení do chodu firmy, jsem byl seznámen s jejími potřebami a zasazen do pozice vývojáře. Postupně jsem řešil danou problematiku. Svůj postup a pokrok jsem konzultoval pravidelně na firmě. Stejně tak jsem mohl dotazy, vzniklé uprostřed řešení, zasílat prostřednictvím emailu.

3 Seznam úkolů

Při prvních návštěvách došlo k mému zaškolení v oblasti chodu a fungování firmy a stejně tak v oblasti vývojového prostředí. Na pozdější schůzce jsem se dozvěděl, že firma Cathedral Software se rozhodla pro změnu technologie webového rozhraní informačního systému. Tyto doteď využívaly skriptovacího programovacího jazyka PHP. Nově zvolili přechod na webovou prezentaci pomocí programovacího jazyku Python. Byla mi přidělena určitá komponenta tohoto celku.

3.1 Možnosti web rozhraní pro přístup k datům s využitím jazyka Python

Prvním úkolem bylo zjištění možností jazyka Python pro tvorbu web rozhraní pro přístup k datům. Jednalo se tedy o část čistě analytickou.

3.2 Tvorba webového rozhraní pro přístup k datům pomocí frameworku Web2Py

Tento úkol rozvíjel úkol první. Šlo o už samostatnou tvorbu webového rozhraní pomocí frameworku Web2Py a jeho funkcionalit.

3.3 Tvorba webového rozhraní pro přístup k datům pomocí frameworku Django

Posledním úkolem byla tvorba webového rozhraní pro přístup k datům frameworkem Django a následné srovnání dvou nástrojů, které jsem pro tvorbu využil

4 Zvolený postup řešení

4.1 Možnosti web rozhraní pro přístup k datům s využitím jazyka Python

Prvním úkolem byl výzkum. Tedy zjištění většiny možností pro tvorbu web rozhraní pro přístup k datům. Jako první mě samozřejmě napadl webový framework Django, s kterým jsem se letmo potkal při studiu v předmětu Skriptovací programovací jazyky a jejich aplikace. Snažil jsem se ovšem najít i další alternativy.

Důležité bylo provést srovnání těchto nástrojů pro vývoj webových aplikací. Začal jsem tedy pátrat po vzájemném srovnání těchto frameworků. Některé z těchto možností nebyly open source, tudíž pro můj vývoj nepoužitelné, jiné nabízely jen trial verzi, která je samozřejmě nevyužitelná pro dlouhodobější vývoj. U těch pro můj záměr vhodných jsem našel ukázky webových stránek:

- Django - Instagram, Mozilla, New York Times
- Pylons - Reddit, Quora
- TurboGears - Kid Rewards
- Web2Py - SourceForge

Důležité bylo také zjistit názory zkušenějších uživatelů, kteří měli třeba šanci tyto frameworky vyzkoušet. Proto jsem začal pátrat po různých diskusních fórech zabývajících se touto tematikou. Na základě těchto názorů jsem si byl již schopen udělat relativně ucelený obrázek o základních kamenech těchto nástrojů:

- Django
 - používá vlastní ORM
 - je nejrozšířenějším frameworkem využívajícím Python - velké množství dokumentace a diskusních vláken
 - spolupráce s MSSQL pouze experimentální
- Pylons
 - soustřeďuje se na správu stránek
 - neklade důraz na sdílení aplikací
- TurboGears
 - využití SQLAlchemy
 - možnost využití jazyků pro tvorbu template

- Web2Py
 - bez nutnosti instalace
 - menší rozšířenost - méně diskusních vláken
 - rozsáhlá dokumentace
 - projekty obsahují velké množství souborů, které ve výsledku zůstávají nevyužity

Nakonec jsem se rozhodl pro práci s frameworkem Django a to právě kvůli mým předchozím zkušenostem s ním, které by mi měly usnadnit začátky práce v něm. Také jsem se rozhodl pracovat ve frameworku Web2Py, a to pro jeho podrobnou dokumentaci a také funkcionality, které nabízel. Tyto dva nástroje jsem se rozhodl využít a na závěr také porovnat.

4.1.1 Použité technologie

4.1.1.1 Web2Py

Jedná se o open-source webový framework pro vývoj webových aplikací s využitím databází. Je vytvořen v jazyce Python. Jde o tzv. full-stack framework, což znamená, že obsahuje všechny komponenty potřebné k vytvoření plně funkčního webu. Web2Py [2] se soustředí na zabezpečení, proto se automaticky zabývá problémy, které by mohly představovat bezpečnostní riziko – jako například injections. Web2Py nevyžaduje instalaci, běží na jakékoli architektuře, na které běží Python a vývoj, nasazení a údržba mohou být prováděny přes vzdálené nebo lokální webové rozhraní. Výhodou je i zpětná kompatibilita při vývoji nových verzí. Dále framework využívá Python syntaxi pro modely controllery a view, ale modely ani kontrolery, na rozdíl od ostatních Python frameworků, neimportuje, ale rovnou je vykoná.

4.1.1.2 Django

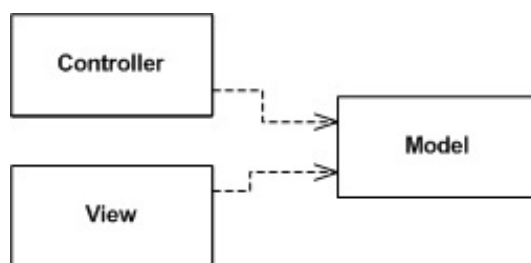
Webový framework, který byl vyvinut v novinářském prostředí a díky tomu se soustředí na rychlý a jednoduchý vývoj. Důležitým principem je tzv. DRY (Don't repeat yourself – neopakovat se), také se zaměřuje na znovupoužitelnost a propojitelnost komponent. Nabízí možnost automatického vygenerování správce aplikace, samostatný server pro testování a vývoj. [3]

4.1.1.3 Model-view-controller

MVC [4] neboli Model-view-controller je softwarová architektura pro tvorbu uživatelských rozhraní. Rozděluje samotnou aplikaci na tři propojené vrstvy, ty oddělují vnitřní reprezentaci informací od způsobu, jakým jsou informace prezentovány nebo přijímány od uživatele.

- model – objekt reprezentující data, reaguje na instrukce pro změnu stavu, reaguje také na žádosti o informace o jeho stavu
- view – má na starosti reprezentaci dat získaných z modelu
- controller – zasílá příkazy modelu, pro aktualizace jeho stavu, zasílá také příkazy view pro změnu reprezentace dat modelu

View a controller tvoří tzv. prezentační část. V aplikaci se nenachází pouze jeden view-controller pár, ale existuje pár pro každý element obrazovky. Tato komponenta je znovupoužitelná, což je důležitou vlastností MVC.



Obrázek 2: Závislosti mezi MVC

Spojení mezi View a Controllerem existuje, ale většina vývojářů jej nevyužívá, proto se závislosti Model-view-controller většinou prezentují diagramem na obrázku č.2. Architekturu MVC alespoň částečně využívají oba dva mnou použité frameworky, a proto bylo nutné ji řádně nastudovat.

4.2 Tvorba webového rozhraní pro přístup k datům pomocí frameworku Web2Py

Vývoje pod operačním systémem Windows v jazyce Python neprobíhají vždy úplně podle představ. Prvním krokem je instalace samotného Pythonu. Ve firmě Cathedral Software probíhá vývoj v prostředí Eclipse. Já osobně jsem s ním měl již také nějaké zkušenosti, a proto jsem si musel nejprve stáhnout PyDev, což je plugin rozšiřující Eclipse o práci v programovacím jazyce Python.

Framework Web2Py má sice své vlastní vývojové prostředí, to jsem si letmo vyzkoušel, ale není kompletně přehledné. Dá se říct, že je přehnaně interaktivní, např. při najetí kurzorem do strany neustále vyskakují panely pro přesměrování, což práci vůbec neusnadňuje. Proto jsem se rozhodl zůstat v prostředí Eclipse. Bylo tak třeba stáhnout source verzi tohoto frameworku. Při zakládání PyDev projektu je třeba zahrnout právě složku se zdrojovou verzí Web2Py. Samozřejmě jako interpreter nutno zvolit nainstalovaný Python.

S Web2Py jsem se setkal poprvé, proto jsem spoustu času věnoval čtení oficiálního manuálu, stejně tak jako sledování instruktážních videí přímo od jeho tvůrce. Poté jsem se postupně začal seznamovat s tímto frameworkem. Po několika menších projektech, na kterých jsem si osahal vlastnosti jednotlivých funkcionalit, jsem se pustil do projektu vlastního.

Při spuštění projektu ve vývojovém prostředí Eclipse se objeví Web2Py desktopové GUI, kde si zvolíme server, na kterém si projekt rozjedeme. Stejně tak jako port a heslo pro přístup k administraci. Přes toto GUI server také spustíme. Nyní stačí ve webovém prohlížeči zadat výše zmíněnou adresu.

Web2Py pracuje s Model-view-controller (MVC) softwarovou architekturou, ta odděluje datový model, uživatelské rozhraní a řídicí logiku. Tyto tři komponenty jsou nezávislé, a proto by úprava jedné z nich měla mít minimální dopad na zbylé dvě.

Nejprve jsem si tedy pomocí nástroje na návrh databází Toad Data Modeler vyprojektoval vyhovující databázi. Poté jsem si na Microsoft SQL Server 2008 databázi vytvořil a naplnil vhodnými záznamy. Dalším krokem bylo tuto databázi připojit k mému projektu. K tomu bylo třeba v modelové vrstvě zadat správný connection string. Toto se bohužel neobešlo bez problémů. Web2Py je schopen spolupracovat se rozmanitou škálou databází. Výchozí připojení je k databázi SQLite3. S tímto připojením jsem v počátcích pracoval, ale poté jsem chtěl využít své databáze. Každý typ podporované databáze má vlastní connection string, který je v případě MSSQL:

```
mssql://username:password@localhost/test
```

- mssql - typ databáze na kterou se bude v modelové vrstvě připojovat
- username - uživatelské jméno pro přihlášení do databáze
- password - uživatelské heslo pro přihlášení do databáze
- localhost - server na kterém se databáze nachází, v tomto případě localhost

- test - název databáze uložené na serveru

V tomto okamžiku jsem narazil na problém. MSSQL connection string počítá s SQL Server autentizací, která obsahuje uživatelské jméno a heslo. Můj server se ovšem připojuje pomocí Windows autentizace. Ta ale heslo neobsahuje, tím pádem, tudíž funkce pro připojení obdrží jiný než očekávaný string a nelze se tedy připojit. Při hledání řešení, jsem narazil na dotazy uživatelů, kteří narazili na podobný problém. Bohužel ani jedno z navrhovaných řešení na fórech, nevyhovovalo mému případu, a proto jsem byl nakonec nucen změnit typ připojení. A připojit se pomocí SQL Server autentizace.

Po napojení na databázi je třeba definovat tabulky z MSSQL databáze v modelové vrstvě, a hlavně jejich vzájemné vztahy. Ty se totiž z databáze nepřenášejí. Proto je důležitou součástí správné definování cizích klíčů v tabulkách. V případě chybného definování jsem se dočkal chybové hlášky, týkající se porušení integrity databáze. Po vyřešení tohoto problému bylo nutné nastavit omezení hodnot zadávaných atributů.

Kupříkladu rozsah zadávaných hodnot, popřípadě jejich datový typ nebo různé podmínky jako například, aby se zadávaný atribut již nacházel v databázi. Toto je nutné, aby uživatel zadával do databáze údaje v požadovaném tvaru. Tím jsem vyřešil modelovou vrstvu MVC. Další na řadě byla vrstva řídicí.

V řídicí vrstvě bylo nutné vytvořit funkci pro vytvoření formuláře pro vkládání záznamů do databáze. Web2Py nabízí předdefinovaný konstruktor `SQLFORM`, který vytvoří automaticky formulář pro naplnění tabulek databáze. Tento konstruktor čerpá informace z modelové vrstvy. V té, podle typu zdefinovaných atributů vytvoří příslušnou část formuláře. Toto řešení je vhodné pro vytvoření rychlého řešení, kde nám nezáleží na vzhledu a vlastní úpravě formátu formuláře. Tuto funkcionalitu jsem si vyzkoušel, ale nakonec jsem se rozhodl vytvořit formulář vlastní, abych jej mohl upravovat podle konkrétních požadavků. Formulář obsahoval všechny základní prvky, od `TextBoxu` po `ComboBox`, stejně tak jako prvky lehce pokročilejší a to podmíněná pole, která se objeví jen za dodržení určitých podmínek.

Na přidružených stránkách Web2py jsem našel plugin pro reprezentaci dat na webovém rozhraní. Tento plugin byl označen jako experimentální. Práce s ním nebyla nijak jednoduchá. Podařilo se mi vytvořit funkci pro naplnění gridu daty z mé databáze. Tyto data se dala i zobrazit v detailu. Záznamy bylo možno mazat a přidávat, problémem ovšem byla jejich editace. Provedené změny se v databázi nezaznamenávaly. Tento problém se mi, ani s pomocí diskusních fór věnujících se Web2Py problematice, nepodařilo vyřešit, i když někteří uživatelé měli obdobný problém.

Musel jsem tedy začít hledat jiné řešení. Jako použitelná možnost se jevila funkcionalita přímo zakomponovaná ve Web2Py. Je součástí již zmiňovaného `SQLFORM` konstruktoru a s jeho pomocí jde vytvořit jednoduchý grid. Při definování lze nastavit mnoho parametrů – nejdůležitější z nich jsou:

- `fields` - určuje seznam atributů, které budou z databáze vyzvednuty, také určuje které atributy budou zobrazeny na webovém rozhraní

- headers - slovník, který přiřazuje atributům z tabulky názvy, které se zobrazí na webovém rozhraní
- orderby - nastavuje výchozí atribut, podle kterého dojde k seřazení řad
- paginate - nastavuje maximální počet řad na stránku

Tento konstruktor sestaví podle zadaných parametrů tabulku, kde můžeme záznamy jednoduše setřídít podle námi zvolených atributů, stejně tak jako záznamy do tabulky vkládat, upravovat je a mazat.

Poslední vrstva, kterou jsem musel řešit, byla vrstva reprezentující uživatelské rozhraní – tedy webová prezentace. Web2Py nám opět nabídne předdefinovanou šablonu, kterou jsem se rozhodl nevyužít a pustil jsem se do tvorby vlastního vzhledu. Je možno si vytvořit HTML vzhled, do něhož můžeme vkládat další šablony pro jednotlivé stránky, ale rozhodl jsem se pro tvorbu každé stránky zvlášť. Základem bylo vytvoření přehledného menu, které slouží k přepínání mezi formuláři a zobrazením rozhraní pro přístup k datům. Při tvorbě této webové prezentace jsem své základní znalosti v oblasti HTML [5], rozšířil o znalosti týkající se tvorby navigačního menu.

Výsledný produkt byl využitelný, bohužel nevýhodou řešení pomocí SQLFORM je malá pravomoc v oblasti vzhledu vygenerované tabulky, stejně tak jako v úpravě funkcionality předdefinovaných komponent. Toto řešení je proto použitelné, není však ideální.

Samotná práce s Web2Py byla dosti záludná. Začátky byly značně komplikované a spoustu času trvala adaptace na přítomnost takového množství komponent. Dále učení práce s nimi zabralo také nebylo nejjednodušší. Postupem času jsem si ovšem na přítomnost těchto nástrojů zvykl a v závěru mé práce už jsem je využíval bez sebemenších problémů a opravdu mi nějakou práci usnadnily

4.3 Tvorba webového rozhraní pro přístup k datům pomocí frameworku Django

Mým třetím a posledním úkolem bylo vytvoření webového rozhraní pro přístup k datům pomocí frameworku Django. Instalace tohoto nástroje do vývojového prostředí Eclipse bylo ve srovnání s Web2Py jednodušší. Nutné je pouze provést samotnou instalaci Django a poté už jen zvolit při vytváření nového projektu Pydev Django Project, v případě již existujícího projektu stačí tento projekt nastavit jako Django Project.

Po vytvoření nového projektu se v příslušném adresáři vygenerují následující soubory

- `manage.py` - správa projektu – základní struktura nové aplikace, databázové schéma, testovací webový server
- `settings.py` - konfigurace databázového připojení
- `urls.py` - mapuje url na jednotlivá views
- `views.py` - obsahuje jednotlivé funkce pro views
- `models.py` - datový model aplikace
- `__init__.py` - z aplikace dělá balíček Python

Nejdříve je nutné v `settings.py` nakonfigurovat připojení. V mém případě jsem pracoval s databází SQLite3. Zvolil jsem tedy typ připojení, název databáze a další příslušnosti jako například uživatelské jméno a heslo. V souboru `models.py` bylo nutné nadefinovat datový model celé aplikace. Důležitým krokem byl import funkce `models`, která je v Django již předdefinovaná a umožňuje tvorbu tabulek. Bohužel se syntaxe výrazně lišila od Web2Py syntaxe, proto jsem musel celý model vytvořit znovu. Opět bylo nutné dávat si pozor na předávání cizích klíčů mezi tabulkami.

Poté bylo nutné vytvořit funkce pro přidání položky, její editaci a následné uložení a smazání. Tyto funkce jsem vytvořil v souboru `views.py`. Pro tyto funkce jsem do souboru prvně importoval pomocné Django balíčky pro usnadnění. Využil jsem například hojně funkci `render_to_response` z balíčku `django.shortcuts`, díky kterému bylo možno snadno přesměrovávat z jednoho URL na jiné a také zasílat proměnné. Dále je do souboru nutné importovat nadefinované tabulky z `models.py`, aby s nimi výše zmíněné funkce mohly pracovat.

Následovala část týkající se webové prezentace. Zde jsem se musel naučit pracovat s CSS styly, s kterými jsem se do té doby nikdy nesetkal. Pomohly mi celou stránku rozdělit do menších částí, kde jsem vložil nutné informace a také tabulku pro přístup k datům. Zde bylo nutné propojit jednotlivá pole s daty z `models.py`. Reprezentace dat by nám byla samozřejmě k ničemu, kdybychom žádná data nemohli vložit, a proto jsem pro příslušné tabulky vytvořil formy pro vkládání.

Po konzultaci na firmě bylo mým posledním krokem zobrazení mnou vytvořených funkcí, aby nebylo nutné přepisovat kód na více místech. Tato vytvořená komponenta lze upravovat libovolně podle požadavků, ať už jde o vzhled nebo o změny ve funkčnosti. Dále by se dala jistě i dále rozšiřovat a vyvíjet, je tedy vhodná pro další užití.

4.3.1 Srovnání technologií

Během odborné praxe jsem měl dost času a příležitostí se seznámit s oběma frameworky. Každý měl své výhody, ale zároveň i svá úskalí. Web2Py mi nabídl spoustu komponent, které usnadňovaly některé prvky mé práce, na druhou stranu jen seznámení s funkčností těchto nástrojů mi zabralo spoustu času a i přesto si myslím, že jsem poznal jen zlomek toho, co by se pomocí těchto komponent dalo zvládnout. U Django jsem využil jen pár základních funkcí z balíků, které jsou nainstalovány společně se samotným Djangem, zbylé funkcionality jsem musel dotvořit sám.

Zkušený programátor proto určitě dá přednost Django před Web2Py, protože by ho přídatná komponenta jen zdržovala, začínající programátor by mohl být docela na vážkách, protože samotné učení se práce s těmito nástroji zabere velké množství času. Můj případ budiž toho důkazem, kdy mi práce s jednotlivými frameworky zabrala přibližně stejnou dobu.

5 Uplatněné teoretické a praktické znalosti

V průběhu praxe jsem uplatnil spoustu zkušeností nabytých během mého studia na Vysoké škole báňské – Technické univerzita Ostrava. Z počátku mi pomohlo, že jsem díky předmětu *Skriptovací programovací jazyky a jejich aplikace* měl jisté zkušenosti s programovacím jazykem Python. Toto mi usnadnilo nejen začátky při programování, ale také i výzkumnou část praxe. Ve stejném předmětu jsem se také letmo seznámil s frameworkem Django, takže i tady byla práce s ním lehce usnadněna. V tomto předmětu jsem se také setkal s jazykem HTML, ale spíše jen okrajově.

Dalšími znalostmi, který mi práci částečně usnadnily byly ty, získané z dvojce předmětů, která se zabývá databázemi. Jmenovitě *Úvod do databázových systémů* a *Databázové a informační systémy*. Díky těmto předmětům jsem měl už zkušenosti relačním datovým modelem, návrhem samotné databáze, stejně tak jako s přístupem k datům. Další výhodou byly i zkušenosti s nástroji pro práci s databázemi a základy MSSQL jazyka.

V několika předmětech jsem se také setkal s využitím softwarové architektury Model-view-controller, na jejímž principu pracuje většina frameworků pro kompletní vytvoření webové aplikace. Díky tomu mi nebyla funkčnost těchto nástrojů úplně cizí.

6 Scházející znalosti a dovednosti

Během praxe jsem musel samozřejmě určité věci sám dostudovat. S prací s frameworkem Web2Py jsem neměl žádné zkušenosti, proto jsem byl nucen spoustu času věnovat zjišťování různých funkcionalit tohoto nástroje. Stejně tak bylo nutné se naučit pracovat s frameworkem Django.

Dále jsem neměl moc zkušeností s jazykem HTML, kde jsem znal jen základní syntaxi, a proto jsem musel studovat i materiály týkající se tohoto tématu. Stejně tak jsem při tvorbě webové prezentace neměl žádné zkušenosti s jazykem CSS, proto jsem se práci s ním musel naučit od úplných základů.

7 Závěr

Odbornou praxi jsem si zvolil, abych zjistil, jak to ve firmě zabývající se informačními technologiemi funguje. Přece jen jde o práci, které bych se chtěl v budoucnu věnovat. To je podle mě jedna z nejdůležitějších věcí, kterou bych se při zpracování klasické bakalářské práce nedozvěděl. Proto tato praxe byla pro mě velkou zkušeností.

Naučil jsem se, že důležitou součástí vývoje je výzkum, který se nesmí podcenit, proto jsem mu věnoval spoustu času. Další novou zkušeností byl samotný vývoj, kdy jsem byl sám překvapen, kolik času bylo nutno věnovat jen hledání různých řešení a odpovědí na vyskytnuté problémy na internetu. Novinkou pro mě bylo i přizpůsobování výsledků mé práce konkrétním požadavkům ze strany firmy.

Věřím, že odborná praxe mě obohatila o spoustu zkušeností, které v budoucnu zúročím a rozšířím.

8 Reference

- [1] Internetové stránky firmy Cathedral Software s.r.o. *Cathedral Software* [online]. 2008-2014 [cit. 2014-05-05]. Dostupné z: <http://www.cathedral.cz/>
- [2] Internetové stránky frameworku Web2py. *Web2Py* [online]. 2014 [cit. 2014-05-05]. Dostupné z: <http://web2py.com/book>
- [3] Internetové stránky frameworku Django. *Django* [online]. 2005-2014 [cit. 2014-05-05]. Dostupné z: <https://www.djangoproject.com/>
- [4] Internetové stránky popisující architekturu MVC. *GUI Architectures* [online]. 2014 [cit. 2014-05-05]. Dostupné z: <http://martinfowler.com/eaaDev/uiArchs.html>
- [5] Internetové stránky o HTML a CSS. *HTML Tutorial* [online]. 1999-2014 [cit. 2014-05-05]. Dostupné z: <http://www.w3schools.com/html/>